

Reference 3: JP 2001-346216 A

Paragraphs 0029 to 0034 of JP2001-346216A correspond to paragraphs 0057 to 0069 of US2002/0118756A1.

THIS PAGE BLANK (USPTO)

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2001-346216
(43)Date of publication of application : 14.12.2001

(51)Int.Cl. H04N 7/32
H03M 7/30
H04N 5/92

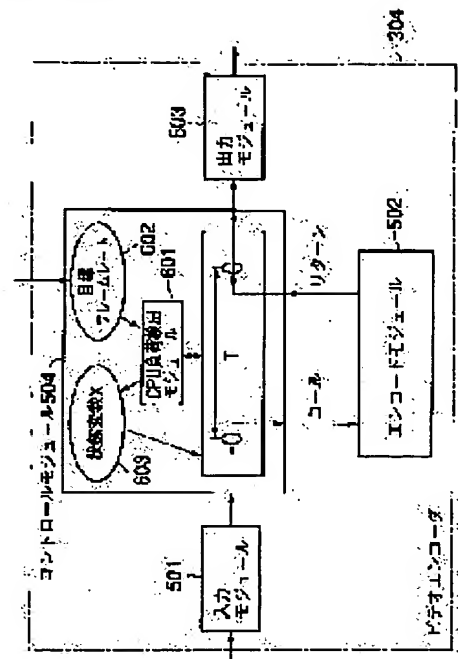
(21)Application number : 2000-168992 (71)Applicant : TOSHIBA CORP
(22)Date of filing : 06.06.2000 (72)Inventor : NAKAMURA SEIICHI
UMEDA SHIGEYUKI

(54) MOVING PICTURE COMPRESSION METHOD AND INFORMATION PROCESSOR

(57)Abstract:

PROBLEM TO BE SOLVED: To exclude occurrence of de-framing in a system where a moving picture is subjected to compression coding by software.

SOLUTION: In the case that a load of a CPU at present is comparatively high and compression coding processing has a delay with respect to a value of an object frame rate 602, the value of a state variable (X) 603 is incremented. Conversely when the current load of the CPU is comparatively low and many more CPU resources can be assigned to the compression coding processing, the value of the state variable (X) 603 is decremented. A parameter of a coded option to be designated is decided for an encode module 502 on the basis of the value of the state variable (X) 603. Thus, while maintaining an object frame rate, the processing contents of the compression coding processing can be optimized in response to the current CPU load.



LEGAL STATUS

[Date of request for examination] 06.06.2000
[Date of sending the examiner's decision of rejection] 27.05.2003
[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]
[Date of final disposal for application]
[Patent number]
[Date of registration]
[Number of appeal against examiner's decision of rejection]
[Date of requesting appeal against examiner's decision of rejection]
[Date of extinction of right]

THIS PAGE BLANK (USPTO)

(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開2001-346216

(P2001-346216A)

(43)公開日 平成13年12月14日(2001. 12. 14)

(51)Int.Cl. ⁷	識別記号	F I	テ-マコ-ト*(参考)
H 0 4 N 7/32		H 0 3 M 7/30	Z 5 C 0 5 3
H 0 3 M 7/30		H 0 4 N 7/137	Z 5 C 0 5 9
H 0 4 N 5/92		5/92	H 5 J 0 6 4

審査請求 有 請求項の数15 O L (全 12 頁)

(21)出願番号 特願2000-168992(P2000-168992)

(22)出願日 平成12年6月6日(2000. 6. 6)

(71)出願人 000003078

株式会社東芝

東京都港区芝浦一丁目1番1号

(72)発明者 中村 誠一

東京都青梅市末広町2丁目9番地 株式会
社東芝青梅工場内

(72)発明者 梅田 茂之

東京都青梅市末広町2丁目9番地 株式会
社東芝青梅工場内

(74)代理人 100058479

弁理士 鈴江 武彦 (外6名)

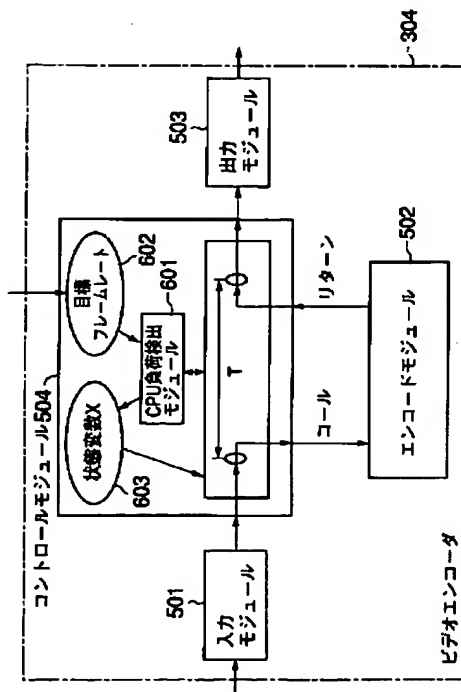
最終頁に続く

(54)【発明の名称】 動画像圧縮方法および情報処理装置

(57)【要約】

【課題】ソフトウェアによって動画の圧縮符号化を行うシステムにおいて、コマ落ちの発生を極力排除する。

【解決手段】現在のCPUの負荷が比較的高く、目標フレームレート602の値に対して圧縮符号化処理に遅れが生じている場合には、状態変数(X)603の値が増分される。逆に、現在のCPU負荷が比較的低く、圧縮符号化処理に対してより多くのCPU資源を割り当てることが可能な場合には、状態変数(X)603の値が減分される。この状態変数(X)603の値に基づき、エンコードモジュール502に対して指定すべき符号化オプションのパラメタが決定される。これにより、目標フレームレートを維持しつつ、現在のCPU負荷に応じて圧縮符号化処理の処理内容を最適化することが可能となる。



【特許請求の範囲】

【請求項 1】 動画像信号の圧縮符号化を行うための動画像圧縮方法であって、
動画像信号を入力するステップと、
入力された動画像信号に対して圧縮符号化処理を施すステップと、

所定のフレームレートが維持されるように、前記圧縮符号化処理の処理状況に基づいて前記圧縮符号化処理の処理内容を制御する制御ステップとを具備することを特徴とする動画像圧縮方法。

【請求項 2】 前記制御ステップは、前記圧縮符号化処理に要した処理時間を検出するステップを含み、検出した処理時間と前記所定フレームレートとに基づいて、前記圧縮符号化処理の処理内容を制御することを特徴とする請求項 1 記載の動画像圧縮方法。

【請求項 3】 前記制御ステップは、前記圧縮符号化処理で行われる動きベクトル検出処理の処理内容を制御することを特徴とする請求項 1 記載の動画像圧縮方法。

【請求項 4】 前記制御ステップは、前記動きベクトル検出処理で行われるブロック探索処理の探索範囲または探索精度を可変制御することを特徴とする請求項 3 記載の動画像圧縮方法。

【請求項 5】 前記入力された動画像信号に対して圧縮符号化処理を施す前に、前記入力された動画像信号に対してフィルタ処理を施すステップをさらに具備し、
前記制御ステップは、前記所定のフレームレートが維持されるように、前記圧縮符号化処理の処理状況に基づいて前記フィルタ処理を制御するステップを含むことを特徴とする請求項 1 記載の動画像圧縮方法。

【請求項 6】 前記圧縮符号化処理は、その圧縮符号化処理のためのコンピュータプログラムを情報処理装置の CPU に実行させることによって行われるものであり、
前記制御ステップは、前記 CPU の負荷状況を検出するステップを含み、前記 CPU の負荷状況に基づいて前記 CPU に実行させる圧縮符号化処理の処理内容を制御することを特徴とする請求項 1 記載の動画像圧縮方法。

【請求項 7】 動画像信号の圧縮符号化を行うための動画像圧縮方法であって、
動画像信号を入力するステップと、

入力された動画像信号に対してフィルタ処理を施すステップと、
フィルタ処理された動画像信号に対して圧縮符号化処理を施すステップと、

所定のフレームレートが維持されるように、前記圧縮符号化処理の処理状況に基づいて前記フィルタ処理および前記圧縮符号化処理の少なくとも一方を制御する制御ステップとを具備することを特徴とする動画像圧縮方法。

【請求項 8】 動画像信号の圧縮符号化を行うことが可能な情報処理装置であって、
動画像信号を入力する入力手段と、

前記入力手段によって入力された動画像信号に対して圧縮符号化処理を施す圧縮処理手段と、
所定のフレームレートが維持されるように、前記圧縮処理手段に実行させる前記圧縮符号化処理の処理内容を制御する制御手段とを具備することを特徴とする情報処理装置。

【請求項 9】 前記制御手段は、前記圧縮符号化処理に要した処理時間を検出する手段を含み、その検出した処理時間と前記所定フレームレートとに基づいて、前記圧縮符号化処理の処理内容を制御することを特徴とする請求項 8 記載の情報処理装置。

【請求項 10】 前記制御手段は、前記圧縮符号化処理で行われる動きベクトル検出処理の処理内容を制御することを特徴とする請求項 8 記載の情報処理装置。

【請求項 11】 前記圧縮符号化処理は、その圧縮符号化処理のためのコンピュータプログラムを前記情報処理装置の CPU に実行させることによって行われるものであり、

前記制御手段は、前記 CPU の負荷状況を検出する手段を含み、前記 CPU の負荷状況に基づいて前記 CPU に実行させる圧縮符号化処理の処理内容を制御することを特徴とする請求項 8 記載の情報処理装置。

【請求項 12】 前記入力された動画像信号に対して圧縮符号化処理が施される前に、前記入力された動画像信号に対してフィルタ処理を施す手段をさらに具備し、
前記制御手段は、前記所定のフレームレートが維持されるように、前記フィルタ処理を制御する手段を含むことを特徴とする請求項 8 記載の情報処理装置。

【請求項 13】 動画像信号の圧縮符号化を行うためのコンピュータプログラムが記録されたコンピュータ読み取り可能な記録媒体であって、
前記コンピュータプログラムは、
動画像信号を入力するステップと、
入力された動画像信号に対して圧縮符号化処理を施すステップと、

所定のフレームレートが維持されるように、前記圧縮符号化処理の処理内容を制御する制御ステップとを具備することを特徴とする記録媒体。

【請求項 14】 前記制御ステップは、前記コンピュータプログラムが実行されるコンピュータの CPU の負荷を検出するステップを含み、前記 CPU の負荷状況に基づいて前記圧縮符号化処理の処理内容を制御することを特徴とする請求項 13 記載の記録媒体。

【請求項 15】 前記コンピュータプログラムは、前記入力された動画像信号に対して圧縮符号化処理が施される前に、前記入力された動画像信号に対してフィルタ処理を施すステップをさらに具備し、
前記制御ステップは、前記所定のフレームレートが維持されるように、前記フィルタ処理を制御するステップを含むことを特徴とする請求項 13 記載の記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は動画像信号の圧縮符号化を行うための動画像圧縮方法および情報処理装置に関し、特に滑らかな動画再生を行えるように改善された動画像圧縮方法および情報処理装置に関する。

【0002】

【従来の技術】近年、コンピュータ技術の発達に伴い、デジタルビデオプレーヤ、セットトップボックス、デジタルTV、デジタルVCR、パーソナルコンピュータ等のマルチメディア対応の各種電子機器が開発されている。この種の電子機器では、MPEG2/MPEG4などの動画像高能率符号化方式で圧縮符号化された動画データを復号・再生するためのデコーダのみならず、最近では、動画データをMPEG2/MPEG4などで圧縮符号化するためのエンコーダも搭載され始めている。

【0003】MPEG2/MPEG4の規格では、動きベクトル検出（動き推定ME: Motion Estimation）による動き補償予測技術が用いられている。動きベクトル検出では、入力フレーム内の注目ブロック毎に参照フレームの中から最も近似するブロックを探索するというブロック探索処理が行われる。探索されたブロックと注目ブロックとの空間的なずれ量が動きベクトルとして求められる。そして、この動きベクトルを基に参照フレームから入力フレームの画像が予測され、その予測画像と入力画像との誤差信号に対して直交変換、量子化、可変長符号化処理が施される。この動き補償予測技術により、高能率の圧縮符号化が実現されている。

【0004】

【発明が解決しようとする課題】しかし、動きベクトル検出のためのブロック探索処理には多くの演算量を必要とするため、動きベクトル検出に時間がかかり、結果的に符号化処理の遅れが引き起こされる場合がある。特に、パーソナルコンピュータなどでソフトウェアエンコードを行う場合には、動きベクトル検出のために大きなプロセッサ負荷がかかり、結果として圧縮符号化処理が、入力される動画像のスピードに間に合わなくなる場合がある。圧縮符号化処理の遅れは、音声と同期して動画像の圧縮符号化をソフトウェアによって行う場合に特に顕在化され、音声とのずれや、入力フレームのコマ落ちなどの問題が生じることになる。この場合、目標フレームレートでの符号化ができなくなるので、その符号化データを復号・再生するとギクシャクとした画像となってしまう。

【0005】本発明は上述の事情に鑑みてなされたものであり、圧縮符号化処理時におけるコマ落ちの発生を極力排除できるようにし、再生時に滑らかな動画再生を行うことが可能な動画像圧縮方法および情報処理装置を提供することを目的とする。

【0006】

【課題を解決するための手段】上述の課題を解決するため、本発明は、動画像信号の圧縮符号化を行うための動画像圧縮方法であって、動画像信号を入力するステップと、入力された動画像信号に対して圧縮符号化処理を施すステップと、所定のフレームレートが維持されるように、前記圧縮符号化処理の処理状況に基づいて前記圧縮符号化処理の処理内容を制御する制御ステップとを具備することを特徴とする。

10 【0007】この動画像圧縮方法においては、圧縮符号化処理の処理状況に基づいて圧縮符号化処理の処理内容が制御され、現在の圧縮符号化処理の処理状況に最適な圧縮符号化処理を行うことができる。したがって、プロセッサ負荷が大きく、圧縮符号化処理に遅れが生じている場合には演算処理の少ない圧縮符号化処理に変更することにより、所定の目標フレームレートを維持することが可能となる。逆にプロセッサに余裕がある場合には、演算処理量がより多い高効率の圧縮符号化処理に切り替えることにより、圧縮率の向上を図ることができる。

20 【0008】圧縮符号化処理の処理状況については、圧縮符号化処理に要した処理時間を検出し、その検出した処理時間と目標とする所定のフレームレートとから判断することができる。圧縮符号化処理に要する処理時間はプロセッサの負荷状況によって変わるため、圧縮符号化処理に要した処理時間を検出することは、プロセッサ負荷を検出していることになる。

30 【0009】圧縮符号化処理では動きベクトル検出処理に多くの時間が費やされるので、圧縮符号化処理で行われる動きベクトル検出処理の処理内容を制御することが好ましい。具体的には、動きベクトル検出処理で行われるブロック探索処理の探索範囲または探索精度を可変制御することにより、圧縮符号化処理の内容を最適化することが可能となる。

40 【0010】また、圧縮符号化処理の前に、動画像信号に対してフィルタ処理を施す場合には、そのフィルタ処理を現在の圧縮符号化処理の処理状況に基づいて制御してもよい。例えばフィルタ処理をスキップさせたり、あるいはそのフィルタ処理の種類を変えることにより、容易に目標とする所定のフレームレートを維持することができる。このようなフィルタ処理の制御は、圧縮符号化処理の制御と組み合わせて利用することはもちろんであるが、どちらか一方のみを制御するようにしても良い。

【0011】

50 【発明の実施の形態】以下、図面を参照して本発明の実施形態を説明する。図1には、本発明の第1実施形態に係る情報処理装置の一例として、パーソナルコンピュータを用いた場合の構成が示されている。このパーソナルコンピュータはノートブックタイプの携帯型のコンピュータシステムであり、ビデオカメラなどから入力された

動画像信号をMPEG2/MPEG4等によって圧縮符号化することができる。

【0012】図1のコンピュータシステムにおいては、図示のように、CPU11、ホスト-PCIブリッジ12、主メモリ13、表示コントローラ14、サウンドコントローラ15、通信インターフェース16、I/Oコントローラ17、PCI-ISAブリッジ18、カメラ20、ハードディスクドライブ(HDD)21、DVDドライブ22などが設けられている。

【0013】CPU11は本システム全体を制御するためのプロセッサであり、主メモリ13にロードされたオペレーティングシステムや他の各種プログラムを実行する。本実施形態においては、動画像信号を圧縮符号化するためのプログラムとして、動画圧縮ソフトウェア100が用いられる。この動画圧縮ソフトウェア100は、入力動画像信号からMPEG2、MPEG4などの各種圧縮形式のAV(オーディオ・ビデオ)ストリームを生成することができ、そのAVストリームを符号化ファイルとしてストレージに記録することができる。

【0014】ホスト-PCIブリッジ12はCPUバス1とPCIバス2を接続するバスブリッジであり、ここには主メモリ13を制御するためのメモリコントロールロジックも内蔵されている。表示コントローラ14は本コンピュータシステムのディスプレイモニタとして使用されるLCDや外部CRTディスプレイを制御する。入力動画像信号を再生しながら圧縮符号化する場合には、圧縮符号化対象の入力動画像信号が表示コントローラ14を通じてディスプレイモニタに表示されると共に、それに同期して圧縮符号化処理が実行される。サウンドコントローラ15は音源として用いられるものであり、マイク151およびスピーカ152を通じて各種オーディオデータの入出力を行う。

【0015】通信インターフェース16は、たとえばUSBやIEEE1394などのシリアルインターフェース規格で外部または内蔵のビデオカメラ20と通信するためのものであり、ビデオカメラ20から動画像信号を取り込むことができる。ビデオカメラ20によって撮影された動画像信号は、動画圧縮ソフトウェア100の制御により、MPEG2、MPEG4などの各種圧縮形式にリアルタイムに変換した後にI/Oコントローラ17を介してHDD21、DVDドライブ22、メモリカード23などの各種記録メディアに記録することができる。

【0016】PCI-ISAブリッジ18はPCIバス2とISAバス3との間を接続するバスブリッジであり、ここにはリアルタイムクロック(RTC)181などの各種システムデバイスが内蔵されている。リアルタイムクロック(RTC)181は時計モジュールであり、オペレーティングシステムによる時間管理に利用される。

【0017】(動画圧縮ソフト)ここで、図2および図3を参照して、動画圧縮ソフトウェア100の基本機能について説明する。

【0018】動画圧縮ソフトウェア100は、カメラ20およびマイク151からそれぞれ入力される動画像信号および音声信号に対して圧縮符号化処理を施し、それによって得たAV符号化ファイルをHDD21、DVDドライブ22、またはメモリカード23などの蓄積メディアに記録することができる。図2には、HDD21にAV符号化ファイル記録する場合の様子が示されている。このAV符号化ファイルは、動画像信号と音声信号をそれぞれデジタル圧縮符号化した後にそれらの符号化ビットストリームを多重化することによって生成されたものである。

【0019】動画圧縮ソフトウェア100は、図3に示されているように、アプリケーションプログラム101と圧縮エンジン102とから構成されている。アプリケーションプログラム101は動画圧縮処理のためのユーザインターフェースと圧縮エンジン102を制御するためのインターフェースとを有しており、ユーザから指定された動画像信号をキャプチャリングおよび圧縮符号化するために必要な動作を、圧縮エンジン102に対して指示する。圧縮エンジン102は、オペレーティングシステム(OS)103上に実装されたマルチメディア処理用のプラットフォームであり、ファイル入出力処理や動画/音声読み込みのためのキャプチャリング処理をはじめとする、動画・音声圧縮のために必要な処理を行う種々のプログラムモジュール(フィルタ)群から構成されている。これらモジュールはアプリケーションプログラム101からの指示により任意に組み合わせて用いることができる。

【0020】(圧縮エンジン)次に、図4を参照して、圧縮エンジン102の機能構成を説明する。圧縮エンジン102は、図示のように、マネージャ301、カメラキャプチャ302、オーディオキャプチャ303、ビデオエンコーダ304、オーディオエンコーダ305、マルチプレクサ306、及びファイルライタ307によって実現される。カメラキャプチャ302、オーディオキャプチャ303、ビデオエンコーダ304、オーディオエンコーダ305、マルチプレクサ306、及びファイルライタ307の動作は、マネージャ301によって管理される。

【0021】まず、カメラキャプチャ302およびオーディオキャプチャ303により、圧縮符号化対象の動画像信号及び音声信号がカメラ20およびマイク151からそれぞれ取り込まれる。カメラキャプチャ302によってキャプチャリングされた動画像信号はビデオエンコーダ304に送られ、そこでキャプチャリングと並行してリアルタイムに動画像の圧縮符号化処理が実行される。また、オーディオキャプチャ303によってキャプ

チャリングされた音声信号はオーディオエンコーダ305に送られ、そこでリアルタイムに音声の圧縮符号化処理が行われる。

【0022】動画画像の圧縮符号化処理では、動きベクトル検出(ME)が行われる。動きベクトル検出(ME)では、入力フレーム内の各注目ブロック毎に参照フレームの中から最も近似するブロックを探索するというブロック探索処理が行われる。探索されたブロックと注目ブロックとの空間的なずれ量が動きベクトルとして求められる。そして、この動きベクトルを基に参照フレームから入力フレームの画像が予測され、その予測画像と入力画像との誤差信号が求められる。この誤差信号に対しては、離散コサイン変換(DCT)、量子化、および可変長符号化処理が施され、これによって符号化ビットストリームが生成される。また、量子化されたブロックに対しては逆量子化、逆DCTが施され、これによりフレーム間予測符号化のための参照フレームが生成される。

【0023】ビデオエンコーダ304によって得られた動画画像の符号化ビットストリーム、およびオーディオエンコーダ305によって得られた音声信号の符号化ビットストリームは、マルチプレクサ306に送られ、そこで多重化される。この多重化された符号化ビットストリームはファイルライタ307に送られ、そこで所定のAV符号化ファイルの形式に変換された後にHDD21等

に書き込まれる。

【0024】(ビデオエンコーダ)次に、図5を参照して、ビデオエンコーダ304の構成を説明する。ビデオエンコーダ304は、図示のように、入力モジュール501、エンコードモジュール502、出力モジュール503、およびコントロールモジュール504から構成されている。

【0025】入力モジュール501はカメラキャプチャ302を介して動画画像信号を入力する。エンコードモジュール502は入力モジュール501によって入力された動画画像信号に対して圧縮符号化処理を施すための機能モジュールであり、コントロールモジュール504の制御の下に入力フレームに対して圧縮符号化処理を行い、その圧縮符号化処理の結果をコントロールモジュール504に返す。出力モジュール503は、エンコードモジュール502によって得られた符号化ビットストリームを前述のマルチプレクサ306に出力する。

【0026】コントロールモジュール504は、ビデオエンコーダ404内の各モジュールの動作を制御するためのものであり、入力フレーム毎にエンコードモジュール502をコールして、その入力フレームに対する圧縮符号化処理をエンコードモジュール502に実行させる。このコントロールモジュール504には、エンコードモジュール502による圧縮符号化処理の処理状況を管理する機能が設けられており、指定された目標フレームレートが少なくとも維持されるように、エンコードモ

ジュール502の圧縮符号化処理の処理状況に基づいてそのエンコードモジュール502に実行させる圧縮符号化処理の処理内容が最適化される。圧縮符号化処理の処理内容の制御は、エンコードモジュール502をコールするときににそのエンコードモジュール502に渡す符号化オプションのパラメタを変えることによって行われる。

【0027】コントロールモジュール504は、CPU負荷検出モジュール601、目標フレームレート602、状態変数(X)603を有している。CPU負荷検出モジュール601は、エンコードモジュール502が圧縮符号化処理に要した時間に基づいて現在のCPU11の負荷状況を検出する。すなわち、CPU負荷検出モジュール601は、エンコードモジュール502のコールからリターンまでの経過時間T(=圧縮符号化処理の処理時間)を常に監視しており、その経過時間Tと、アプリケーションプログラム101を通じてユーザにより指定される目標フレームレート602の値とに基づいて圧縮符号化処理の進捗状況、つまり現在のCPU11の負荷状況を判断する。

【0028】現在のCPU11の負荷が比較的高く、目標フレームレート602の値に対して圧縮符号化処理に遅れが生じている場合には、状態変数(X)603の値が増分される。逆に、現在のCPU11の負荷が比較的低く、圧縮符号化処理に対してより多くのCPU資源を割り当てることが可能な場合には、状態変数(X)603の値が減分される。この状態変数(X)603の値に基づき、エンコードモジュール502に対して指定すべき符号化オプションのパラメタが決定される。

【0029】(画像圧縮制御処理)次に、図6のフローチャートを参照して、ビデオエンコーダ304によって行われる動画画像圧縮制御の全体の流れについて説明する。

【0030】コントロールモジュール504は、エンコードモジュール502に入力フレームの圧縮符号化処理を実行させる前に、まず、処理開始時刻(現在時刻)をオペレーティングシステム103から取得する(ステップS221)。次いで、コントロールモジュール504は、エンコードモジュール502をコールして、現在の入力フレームに対する圧縮符号化処理を実行させる(ステップS222)。エンコードモジュール502による1フレーム分の圧縮符号化処理が終了すると、エンコードモジュール502からコントロールモジュール504に制御が戻される。コントロールモジュール504は、このときの処理終了時刻をオペレーティングシステム103から取得して、処理開始時刻との差をとることによって圧縮処理時間(T)を算出する(ステップS223)。

【0031】次いで、コントロールモジュール504は、圧縮処理時間(T)と目標フレームレート602の

値とを比較することによって、圧縮符号化処理の処理状況を判断する(ステップS224)。圧縮処理時間

(T)の値が目標フレームレート602で指定されるフレーム時間間隔よりも大きい場合、つまり圧縮符号化処理が遅れている場合には、コントロールモジュール504は、状態変数(X)603の値を+1更新する(ステップS225)。一方、圧縮処理時間(T)の値が目標フレームレート602で指定されるフレーム時間間隔を下回った場合、つまりCPU11に余裕がある場合には、コントロールモジュール504は、状態変数(X)603の値を-1更新する(ステップS225)。圧縮処理時間(T)の値が目標フレームレート602で指定されるフレーム時間間隔と等しい場合には、状態変数(X)603の更新は行われない。

【0032】(画像データ圧縮処理)次に、図7のフローチャートを参照して、エンコードモジュール502によって行われる圧縮符号化処理の流れについて説明する。

【0033】まず、入力フレームから圧縮対象のブロックを抽出する処理や、探索対象となる参照フレームを準備する処理などが探索前処理として実行される(ステップS251)。次いで、動きベクトル検出処理が開始されるが、この場合、ブロック探索すべき範囲は現在の状態変数(X)603の値に基づいて可変制御される(ステップS252~S255)。この様子を図8に示す。つまり、

1) 状態変数(X)603の値が0よりも大きい場合には、CPU11の負荷を軽減するために、通常よりも狭い探索範囲E_nがコントロールモジュール504によって指定され、それに基づいてブロック探索処理が行われる

2) 状態変数(X)603の値が0である場合には、通常の探索範囲Eがコントロールモジュール504によって指定され、それに基づいてブロック探索処理が行われる

3) 状態変数(X)603の値が0よりも小さい場合には、符号化効率を高めるために、通常よりも広い探索範囲E_wがコントロールモジュール504によって指定され、それに基づいてブロック探索処理が行われる。

【0034】このようブロック探索によって動きベクトルが検出され、そしてその動きベクトルを基に参照フレームから入力フレームの画像が予測され、その予測画像と入力画像との誤差信号が求められる。そして、この誤差信号に対して、離散コサイン変換(DCT)、量子化、および可変長符号化処理が施される(ステップS256)。

【0035】なお、このような探索範囲の可変制御のみならず、その代わりに、それに加えて、ブロックツリー探索や、1ラインスキップによる探索などを利用することもできる。「ブロックツリー探索」では、図9(A)に示すように、まず、探索範囲内の全ブロックをいくつ

かのグループに分け、入力フレーム内の注目ブロックとの比較を各グループの代表ブロック(二重丸で示す)に関してのみ求められる。次いで、最も類似する代表ブロックを含むグループ内の他のブロック(白丸で図示)との比較が行われることにより、注目ブロックに最も類似するブロックが決定される。これにより、比較対象のブロック数を削減でき、処理の軽減を図ることができる。「1ラインスキップ」は隣接する縦方向の2ラインの画素同士の高類似性を利用して、図9(B)に示すように、ブロック間の画素比較を1ライン飛ばしで行うというものである。これにより、比較のための演算処理量を削減できる。

【0036】例えば、状態変数(X)603の値が1の場合には、通常よりも狭い探索範囲E_nを指定し、状態変数(X)603の値が2の場合には、通常よりも狭い探索範囲E_nを指定するのに加え、ブロックツリー探索および1ラインスキップによる探索のいずれか一方を指定し、さらに状態変数(X)603の値が3の場合には、通常よりも狭い探索範囲E_nを指定するのに加え、ブロックツリー探索および1ラインスキップによる探索の双方を指定することなどにより、遅れ量に応じて演算処理量を段階的に削減することもできる。

【0037】(画像圧縮制御処理#2)次に、図10のフローチャートを参照して、ビデオエンコーダ304によって行われる動画圧縮制御の第2の例について説明する。ここでは、状態変数(X)603は“0”と“1”の2値によって管理される。

【0038】コントロールモジュール504は、エンコードモジュール502に入力フレームの圧縮符号化処理を実行させる前に、まず、処理開始時刻(現在時刻)をオペレーティングシステム103から取得する(ステップS231)。次いで、コントロールモジュール504は、エンコードモジュール502をコールして、現在の入力フレームに対する圧縮符号化処理を実行させる(ステップS232)。エンコードモジュール502による1フレーム分の圧縮符号化処理が終了すると、エンコードモジュール502からコントロールモジュール504に制御が戻される。コントロールモジュール504は、このときの処理終了時刻をオペレーティングシステム103から取得して、処理開始時刻との差をとることによって圧縮処理時間(T)を算出する(ステップS233)。

【0039】次いで、コントロールモジュール504は、圧縮処理時間(T)と目標フレームレート602の値とを比較することによって、圧縮符号化処理の処理状況を判断する(ステップS234)。圧縮処理時間(T)の値が目標フレームレート602で指定されるフレーム時間間隔よりも大きい場合、つまり圧縮符号化処理が遅れている場合には、コントロールモジュール504は、状態変数(X)603の値を“1”に設定する

(ステップS235)。一方、圧縮処理時間(T)の値が目標フレームレート602で指定されるフレーム時間間隔と同じ又は下回った場合には、コントロールモジュール504は、状態変数(X)603の値を“0”に設定する(ステップS225)。

【0040】(画像データ圧縮処理#2)次に、図11のフローチャートを参照して、図10の状態変数管理に対応する圧縮符号化処理の制御について説明する。まず、入力フレームから圧縮対象のブロックを抽出する処理や、探索対象となる参照フレームを準備する処理などが探索前処理として実行された後、通常のブロック探索処理が整数画素精度で行われる(ステップS261)。次いで、参照フレーム内の各画素間を一对一で直線補間する処理を伴う0.5画素精度での高精度のブロック探索(半画素探索)が開始されるが、この場合、半画素探索の実行はオプションとして指定され、状態変数(X)603の値に基づいて実行又はスキップされる。

【0041】つまり、状態変数(X)603の値が“0”であれば(ステップS262のYES)、半画素探索処理が実行され(ステップS263)、それによって高精度の動きベクトル検出が行われる。一方、状態変数(X)603の値が“1”であれば(ステップS262のNO)、半画素探索処理はスキップされ、整数画素精度で行われたステップS261のブロック探索結果に基づいて動きベクトルが検出される。

【0042】そしてこの後、検出された動きベクトルを基に参照フレームから入力フレームの画像が予測され、その予測画像と入力画像との誤差信号が求められる。そして、この誤差信号に対して、離散コサイン変換(DCT)、量子化、および可変長符号化処理が施される(ステップS264)。

【0043】(ビデオエンコーダ#2)次に、本発明の第2実施形態について説明する。本第2実施形態は、第1実施形態に比しビデオエンコーダ304の構成だけが異なっており他の点は第1実施形態と同じである。

【0044】図12には、本第2実施形態で用いられるビデオエンコーダ304の構成が示されている。ビデオエンコーダ304は、図示のように、入力モジュール701、プレフィルタモジュール702、エンコードモジュール703、出力モジュール704、およびコントロールモジュール705から構成されている。

【0045】入力モジュール701は図5の入力モジュール501に相当するものであり、カメラキャプチャ302によって取り込まれた動画像信号を入力する。プレフィルタモジュール702は、コントロールモジュール705の制御の下に入力フレームに対してノイズ除去等の画質改善のために平滑化フィルタ処理(高域カット)を行い、フィルタ処理後の入力フレームをコントロールモジュール705に返す。フィルタの種類(IIRフィルタ、FIRフィルタ)の切り換え、およびフィルタパ

ラメータの設定により、処理速度やフィルタ強度の異なる様々なフィルタ処理を圧縮符号化対象の入力フレームに対して施すことができる。

【0046】エンコードモジュール703は図5のエンコードモジュールに相当するものであり、コントロールモジュール705の制御の下に入力フレームに対して圧縮符号化処理を行い、その圧縮符号化処理の結果をコントロールモジュール705に返す。出力モジュール704は、図5の出力モジュール503に対応するものであり、エンコードモジュール703によって得られた符号化ビットストリームを前述のマルチプレクサ306に出力する。

【0047】コントロールモジュール705は、図5のコントロールモジュール504に対応するものである。このコントロールモジュール705には、エンコードモジュール703による圧縮符号化処理の処理状況を管理する機能に加え、プレフィルタモジュール702によるフィルタ処理の処理状況を管理する機能が設けられており、指定された目標フレームレートが少なくとも維持されるように、フィルタ処理の最適化制御が行われる。

【0048】コントロールモジュール705は、CPU負荷検出モジュール801、目標フレームレート802、および2つの状態変数(X, Y)803を有している。CPU負荷検出モジュール801は、プレフィルタモジュール702のフィルタ処理に要した時間を検出するための処理時間モニタ機能と、エンコードモジュール703の圧縮符号化処理に要した時間を検出するための処理時間モニタ機能とを有しており、それら処理時間モニタ機能を用いて現在のCPU11の負荷状況を検出する。すなわち、CPU負荷検出モジュール801は、プレフィルタモジュール702のコールからリターンまでの経過時間 t_1 (=フィルタ処理の処理時間)と、エンコードモジュール703のコールからリターンまでの経過時間 t_2 (=圧縮符号化処理の処理時間)とを常に監視しており、フィルタ処理および圧縮符号化処理のそれぞれについて個別に処理状況を管理する。フィルタ処理の処理状況は状態変数Yによって管理され、また圧縮符号化処理の処理状況は第1実施形態と同様の方法により状態変数Xによって管理される。

【0049】状態変数Yとしては、例えば“0”と“1”の2値の変数が用いられる。状態変数Y=“1”のときは通常通りのフィルタ処理が実行されるが、状態変数Y=“0”のときは演算処理量の少ないフィルタ処理への切り替え、あるいはフィルタ処理のスキップが行われる。

【0050】以下、図13のフローチャートを参照して、図12のビデオエンコーダ304の動作について説明する。ここでは、状態変数Y=“0”のときにフィルタ処理をスキップさせる場合を例示して説明することにする。

【0051】まず、コントローラモジュール705はまず現在の状態変数Yを参照し、状態変数Yが“0”であるか“1”であるかを判断する(ステップS301)。状態変数Y=“0”の場合には、フィルタ処理はスキップされ、エンコードモジュール703によるエンコード処理(ステップS308)に移行する。一方、状態変数Y=“1”の場合には、以下のようなフィルタ処理制御が実行される。

【0052】すなわち、コントロールモジュール705は、プレフィルタモジュール702に入力フレームのフィルタ処理を実行させる前に、まず、処理開始時刻(現在時刻)をオペレーティングシステム103から取得する(ステップS302)。次いで、コントロールモジュール705は、プレフィルタモジュール702をコールして、現在の入力フレームに対するフィルタ処理を実行させる(ステップS303)。プレフィルタモジュール702による1フレーム分のフィルタ処理が終了すると、プレフィルタモジュール702からコントロールモジュール705に制御が戻される。コントロールモジュール705は、このときの処理終了時刻をオペレーティングシステム103から取得して、処理開始時刻との差をとることによってフィルタ処理時間(t_1)を算出する(ステップS304)。

【0053】次いで、コントロールモジュール705は、算出したフィルタ処理時間 t_1 と通常の圧縮符号化処理時間 t_2 とを考慮することにより、動画圧縮全体に要する時間 $T(=t_1+t_2)$ が目標フレームレート802のフレーム時間間隔内に収まるか否かを判断する(ステップS305)。フレーム時間間隔内に収まる場合にはコントロールモジュール705は状態変数Yを“1”に維持するが(ステップS306)、もしフィルタ処理に多くの時間がかかり、動画圧縮全体に要する時間 $T(=t_1+t_2)$ が目標フレームレート802のフレーム時間間隔よりも大きくなるような場合には、コントロールモジュール705は、状態変数Yを“0”に設定する(ステップS307)。

【0054】続く、エンコードモジュール703によるエンコード処理(ステップS308)では、図6および図7で説明した処理、あるいは図10および図11で説明した処理が行われ、例えば、実際の圧縮符号化処理時間 t_2 と、目標フレームレート802のフレーム時間間隔から通常のフィルタ処理時間を差し引いた残りの時間との大小関係に基づいて、圧縮符号化処理に係わる演算処理量の制御(状態変数Xの更新、状態変数Xに基づくブロック探索範囲の可変制御)が行われる。

【0055】この後、コントロールモジュール705は、圧縮符号化処理時間 t_2 に通常のフィルタ処理時間を加算した値が目標フレームレート802のフレーム時間間隔内に収まるか否かを判断し(ステップS309)、収まる場合には状態変数Yを“1”に設定し(ス

テップS310)、収まらない場合には状態変数Yを“0”に設定する(ステップS311)。

【0056】なお、本例ではフィルタ処理の制御と圧縮符号化処理の制御とを組み合わせる場合を説明したが、CPU負荷に応じて、フィルタ処理と圧縮符号化処理のいずれか一方の処理内容のみを制御するようにしても良い。この場合の例を図14に示す。

【0057】まず、動画圧縮全体に要する時間 $T(=t_1+t_2)$ が目標フレームレート802のフレーム時間間隔内であるか調べられる(ステップS401)。CPU負荷が大きく、時間 T がフレーム時間間隔を上回る場合には、例えばアプリケーションプログラム101を通じてユーザから指定される入力動画像信号に関する画質情報等に基づき、ノイズの大きい低画質の画像であるか、ノイズの少ない高画質の画像であるかを判断する(ステップS402)。ノイズの大きい低画質の画像の場合には、フィルタ処理を省くと、圧縮符号化処理後の画質に大きな影響を及ぼすことになるため、フィルタ処理については演算量を減らさずに、圧縮符号化処理の演算処理を低減するための処理のみを行う(ステップS403)。一方、ノイズの少ない高画質の画像であれば、フィルタ処理を行わずとも高画質の符号化画像が得られるので、フィルタ処理の演算量を低減させる処理(フィルタ処理スキップを含む)のみを行う(ステップS404)。

【0058】以上のように、本各実施形態によれば、目標フレームレートを維持しつつ、CPU負荷に応じた圧縮符号化処理(さらにはフィルタ処理)の最適化制御を行うことが可能となる。さらに、本各実施形態のソフトウェアエンコードの方法は、その手順を含むコンピュータプログラムをコンピュータ読み取り可能な記録媒体を通じて通常のコンピュータに導入するだけで、容易に実現することができる。また、マイクロプロセッサベースでエンコードを行うものであれば、コンピュータのみならず、例えばデジタルビデオカメラや、他の各種電子機器に適用しても有効である。

【0059】また、本発明は、上記各実施形態に限定されるものではなく、実施段階ではその要旨を逸脱しない範囲で種々に変形することが可能である。更に、上記実施形態には種々の段階の発明が含まれており、開示される複数の構成要件における適宜な組み合わせにより種々の発明が抽出され得る。例えば、実施形態に示される全構成要件から幾つかの構成要件が削除されても、発明が解決しようとする課題の欄で述べた課題が解決でき、発明の効果の欄で述べられている効果が得られる場合には、この構成要件が削除された構成が発明として抽出され得る。

【0060】

【発明の効果】以上説明したように、本発明によれば、圧縮符号化処理時におけるコマ落ちの発生を極力排除で

きるようになり、再生時に滑らかな動画再生を行うことが可能となる。

【図面の簡単な説明】

【図1】本発明の第1実施形態に係るコンピュータシステムの構成を示すブロック図。

【図2】同第1実施形態で用いられる動画圧縮ソフトの基本機能を説明するための図。

【図3】同第1実施形態で用いられる動画圧縮ソフトの構造を示す図。

【図4】同第1実施形態で用いられる動画圧縮ソフトの圧縮エンジン部の機能構成を示す図。

【図5】同第1実施形態の圧縮エンジン部に設けられるビデオエンコーダの機能構成を示す図。

【図6】同第1実施形態における画像圧縮制御処理の手順の第1の例を示すフローチャート。

【図7】同第1実施形態における画像データ圧縮処理の手順の第1の例を示すフローチャート。

【図8】同第1実施形態で用いられるブロック探索範囲可変制御の様子を示す図。

【図9】同第1実施形態で用いられるブロックツリー探索およびラインスキップを説明するための図。

【図10】同第1実施形態における画像圧縮制御処理の手順の第2の例を示すフローチャート。

【図11】同第1実施形態における画像データ圧縮処理の手順の第2の例を示すフローチャート。

*

*【図12】本発明の第2実施形態で用いられるビデオエンコーダの機能構成を示す図。

【図13】図12のビデオエンコーダの制御処理の手順を説明するためのフローチャート。

【図14】図12のビデオエンコーダの制御処理の手順の他の例を説明するためのフローチャート。

【符号の説明】

11…CPU

100…動画圧縮ソフト

101…動画圧縮用アプリケーションプログラム

102…圧縮エンジン

301…マネージャ

302…カメラキャプチャ

303…オーディオキャプチャ

304…ビデオエンコーダ

305…オーディオエンコーダ

306…マルチプレクサ

307…ファイルライタ

501, 701…入力モジュール

502, 703…エンコードモジュール

503, 704…出力モジュール

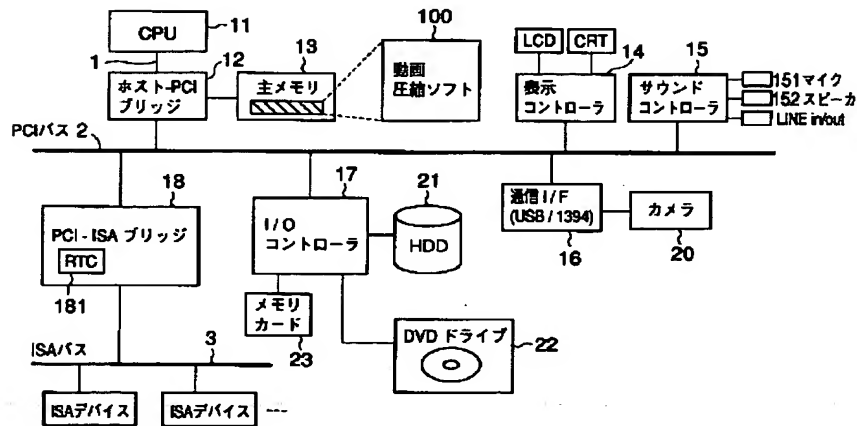
504, 705…コントロールモジュール

601, 801…CPU負荷検出モジュール

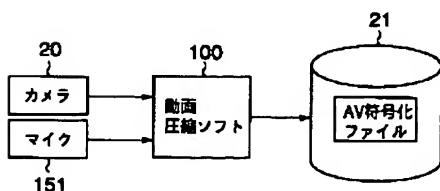
602, 802…目標フレームレート

603, 803…状態変数

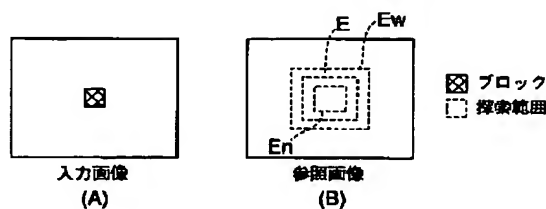
【図1】



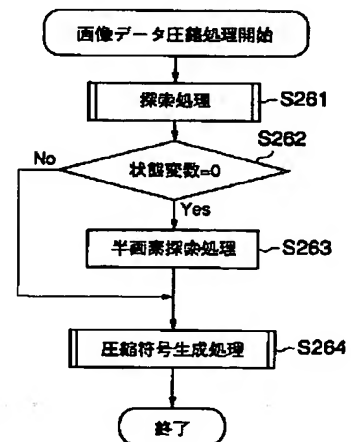
【図2】



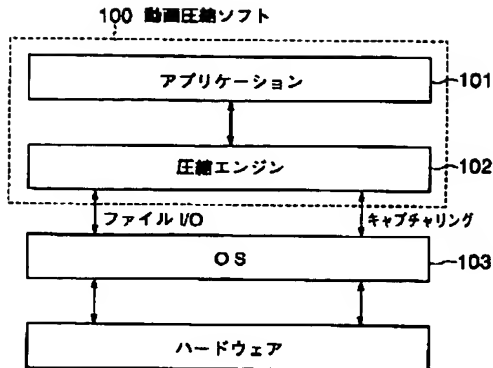
【図8】



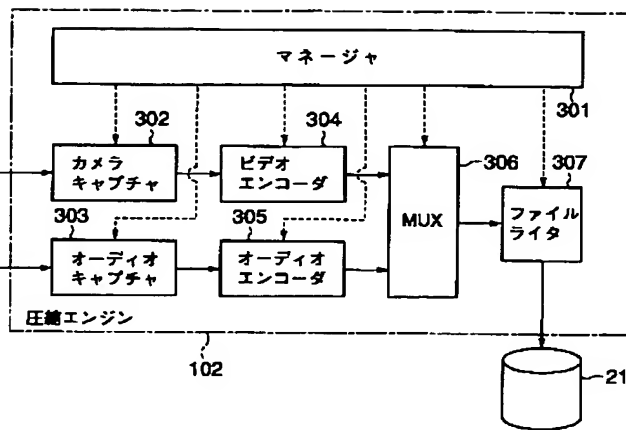
【図11】



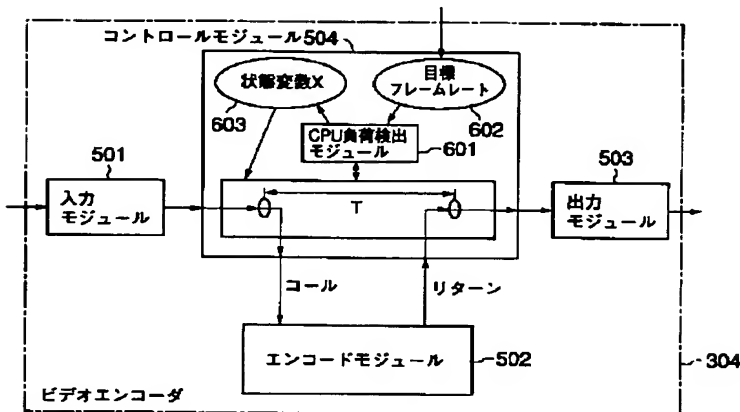
【図3】



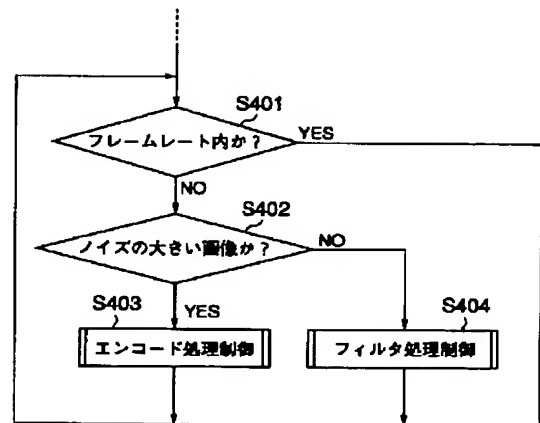
【図4】



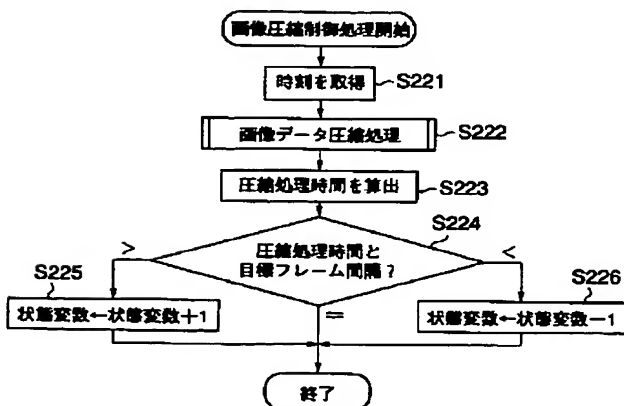
【図5】



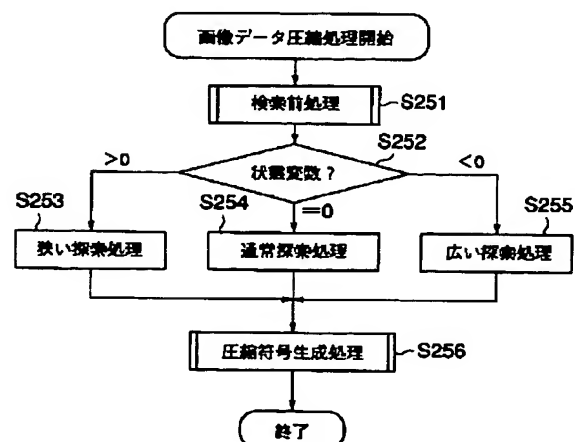
【図14】



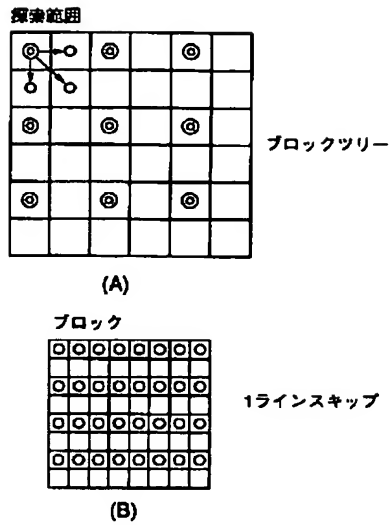
【図6】



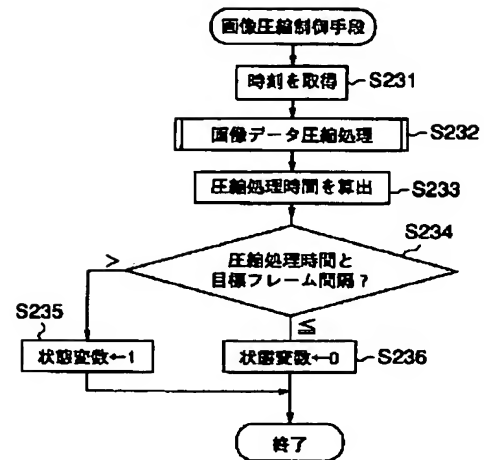
【図7】



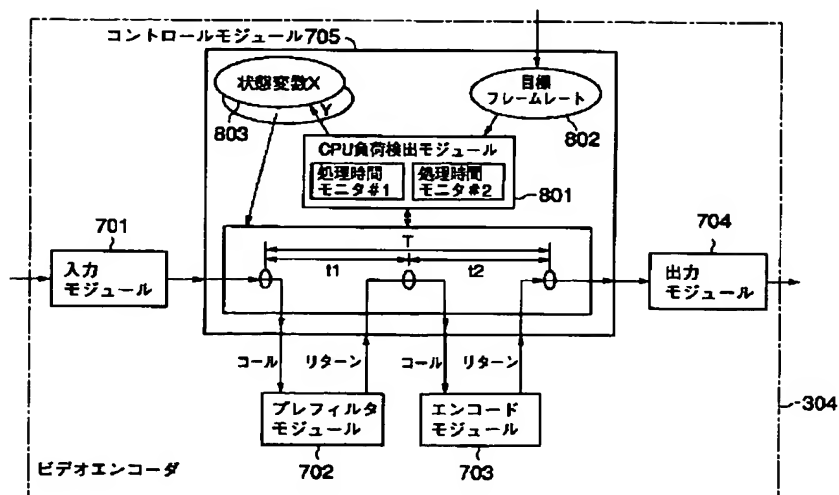
【図9】



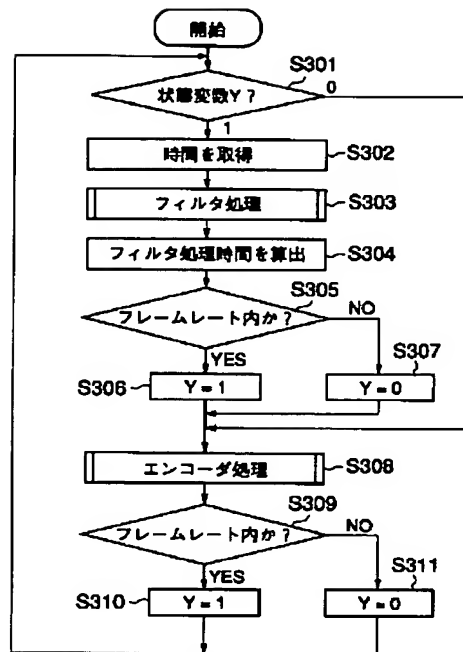
【図10】



【図12】



【図13】



フロントページの続き

F ターム(参考) 5C053 FA20 FA24 FA27 GB07 GB08
 GB17 GB38 JA01 KA05 KA24
 LA01 LA06 LA14
 5C059 KK10 KK22 KK35 MA23 MC11
 ME01 NN01 NN28 PP04 SS02
 SS13 SS26 TA17 TA62 TA63
 TB08 TC00 TC19 UA38 UA39
 5J064 AA01 BA01 BA09 BA16 BB03
 BB12 BC01 BC16 BC26 BD01